

1. KB

- a. Leaf(0)
- b. Leaf(1)
- c. $\forall x(\text{Leaf}(x) \Rightarrow \text{Tree}(x))$
- d. $\forall x \forall y((\text{Tree}(x) \wedge \text{Tree}(y)) \Rightarrow \text{Tree}((x,y)))$
- e. $\forall x(\text{Leaf}(x) \Rightarrow (\text{Size}(x) = 1))$
- f. $\forall x(\exists z) \forall y(\exists w)((\text{Tree}(x) \wedge (\text{Size}(x) = z)) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = w))) \Rightarrow (\text{Size}((x,y)) = z + w))$
- g. $\forall x \forall y(((x = y) \wedge \text{Tree}(x)) \Rightarrow \text{Tree}(y))$
- h. $\forall x \forall y(((\text{Tree}(x) \wedge \text{Tree}(y)) \wedge (x = y)) \Rightarrow (\text{Size}(x) = \text{Size}(y)))$

2. Prove: $\alpha \models (\exists x)(\text{Tree}(x) \wedge (\text{Size}(x) = 1 + 1 + 1))$

Line	Statement	Reason
L 1	Leaf(0)	from KB
L 2	$\forall x(\text{Leaf}(x) \Rightarrow \text{Tree}(x))$	from KB
L 3	Leaf(0) \Rightarrow Tree(0)	from L2 using \forall elimination and x maps to 0
L 4	Tree(0)	From L1 and L3 using Modus Ponens
L 5	$\forall x(\text{Leaf}(x) \Rightarrow (\text{Size}(x) = 1))$	from KB
L 6	Leaf(0) \Rightarrow (Size(0) = 1)	from L5 using \forall elimination and x maps to 0
L 7	Size(0) = 1	from L1 and L6 using Modus Ponens
L 8	$\forall x \forall y((\text{Tree}(x) \wedge \text{Tree}(y)) \Rightarrow \text{Tree}((x,y)))$	from KB
L 9	$\forall y((\text{Tree}(0) \wedge \text{Tree}(y)) \Rightarrow \text{Tree}((0,y)))$	from L8 using \forall elimination and x maps to 0
L 10	$(\text{Tree}(0) \wedge \text{Tree}(0)) \Rightarrow \text{Tree}((0,0))$	from L9 using \forall elimination and y maps to 0
L 11	Tree(0) \wedge Tree(0)	from L4 using AND introduction
L 12	Tree((0,0))	from L10 and L11 using Modus Ponens
L 13	$\forall y((\text{Tree}((0,0)) \wedge \text{Tree}(y)) \Rightarrow \text{Tree}((0,0),y))$	from L8 using \forall elimination and x maps to (0,0)
L 14	$(\text{Tree}((0,0)) \wedge \text{Tree}(0)) \Rightarrow \text{Tree}(((0,0),0))$	from L13 using \forall elimination and y maps to 0
L 15	Tree((0,0)) \wedge Tree(0)	from L4 and L12 using AND introduction

L 16	$\text{Tree}(((0,0),0))$	from L14 and L15 using Modus Ponens
L 17	$\forall x(\exists z \forall y(\exists w)((\text{Tree}(x) \wedge (\text{Size}(x) = z)) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = w))) \Rightarrow (\text{Size}(x,y) = z + w))$	from KB
L 18	$(\exists z) \forall y(\exists w)((\text{Tree}(0) \wedge (\text{Size}(0) = z)) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = w))) \Rightarrow (\text{Size}((0,y)) = z + w)$	from L17 using \forall elimination and x maps to 0
L 19	$\forall y(\exists w)((\text{Tree}(0) \wedge (\text{Size}(0) = 1)) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = w))) \Rightarrow (\text{Size}((0,y)) = 1 + w)$	from L7 and L18 using existential instantiation and z maps to 1
L 20	$(\exists w)((\text{Tree}(0) \wedge (\text{Size}(0) = 1)) \wedge (\text{Tree}(0) \wedge (\text{Size}(0) = w))) \Rightarrow (\text{Size}((0,0)) = 1 + w)$	from L19 using \forall elimination and y maps to 0
L 21	$((\text{Tree}(0) \wedge (\text{Size}(0) = 1)) \wedge (\text{Tree}(0) \wedge (\text{Size}(0) = 1))) \Rightarrow (\text{Size}((0,0)) = 1 + 1)$	from L7 and L20 using existential instantiation and w maps to 1
L 22	$\text{Tree}(0) \wedge (\text{Size}(0) = 1)$	from L4 and L7 using AND introduction
L 23	$(\text{Tree}(0) \wedge (\text{Size}(0) = 1)) \wedge (\text{Tree}(0) \wedge (\text{Size}(0) = 1))$	from L22 using AND introduction
L 24	$\text{Size}((0,0)) = 1 + 1$	from L21 and L23 using Modus Ponens
L 25	$(\exists z) \forall y(\exists w)((\text{Tree}((0,0)) \wedge (\text{Size}((0,0)) = z)) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = w))) \Rightarrow (\text{Size}(((0,0),y)) = z + w)$	from L17 using \forall elimination and x maps to $(0,0)$
L 26	$\forall y(\exists w)((\text{Tree}((0,0)) \wedge (\text{Size}((0,0)) = 1 + 1)) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = w))) \Rightarrow (\text{Size}(((0,0),y)) = 1 + 1 + w)$	from L24 and L25 using existential instantiation and z maps to $1 + 1$
L 27	$(\exists w)((\text{Tree}((0,0)) \wedge (\text{Size}((0,0)) = 1 + 1)) \wedge (\text{Tree}(0) \wedge (\text{Size}(0) = w))) \Rightarrow (\text{Size}(((0,0),0)) = 1 + w)$	from L26 using \forall elimination and y maps to 0
L 28	$((\text{Tree}((0,0)) \wedge (\text{Size}((0,0)) = 1 + 1)) \wedge (\text{Tree}(0) \wedge (\text{Size}(0) = 1))) \Rightarrow (\text{Size}(((0,0),0)) = 1 + 1 + 1)$	from L7 and L27 using existential instantiation and w maps to 1
L 29	$\text{Tree}((0,0)) \wedge (\text{Size}((0,0)) = 1 + 1)$	from L24 and L12 using AND introduction
L 30	$(\text{Tree}((0,0)) \wedge (\text{Size}((0,0)) = 1 + 1)) \wedge (\text{Tree}(0) \wedge (\text{Size}(0) = 1))$	from L29 and L22 using AND introduction
L 31	$\text{Size}(((0,0),0)) = 1 + 1 + 1$	from L28 and L30 using Modus Ponens
L 32	$\text{Tree}(((0,0),0)) \wedge (\text{Size}(((0,0),0)) = 1 + 1 + 1)$	from L16 and L31 using AND introduction
L 33	$(\exists x)(\text{Tree}(x) \wedge (\text{Size}(x) = 1 + 1 + 1))$	from L32 using \exists introduction

3. Skolemize, convert to CNF, and provide a refutation for $KB \cup \neg\alpha$

Line	Statement	Reason
L 1	Leaf(0)	from KB
L 2	Leaf(1)	from KB
L 3	$\forall x(\text{Leaf}(x) \Rightarrow \text{Tree}(x))$	from KB
L 4	$(\text{Leaf}(x) \Rightarrow \text{Tree}(x))$	open formula of L3
L 5	$\forall x \forall y((\text{Tree}(x) \wedge \text{Tree}(y)) \Rightarrow \text{Tree}((x,y)))$	from KB
L 6	$((\text{Tree}(x) \wedge \text{Tree}(y)) \Rightarrow \text{Tree}((x,y)))$	open formula of L5
L 7	$\forall x(\text{Leaf}(x) \Rightarrow (\text{Size}(x) = 1))$	from KB
L 8	$(\text{Leaf}(x) \Rightarrow (\text{Size}(x) = 1))$	open formula of L7
L 9	$\forall x(\exists z) \forall y(\exists w)((\text{Tree}(x) \wedge (\text{Size}(x) = z)) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = w))) \Rightarrow (\text{Size}((x,y)) = z + w)$	from KB
L 10	$\forall x \forall y(((\text{Tree}(x) \wedge (\text{Size}(x) = f_1(x,y))) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = f_2(x,y)))) \Rightarrow (\text{Size}((x,y)) = f_1(x,y) + f_2(x,y)))$	Skolemization of L9
L 11	$((\text{Tree}(x) \wedge (\text{Size}(x) = f_1(x,y))) \wedge (\text{Tree}(y) \wedge (\text{Size}(y) = f_2(x,y)))) \Rightarrow (\text{Size}((x,y)) = f_1(x,y) + f_2(x,y))$	open formula of L10
L 12	$\forall x \forall y(((x = y) \wedge \text{Tree}(x)) \Rightarrow \text{Tree}(y))$	from KB
L 13	$((x = y) \wedge \text{Tree}(x)) \Rightarrow \text{Tree}(y)$	open formula of L12
L 14	$\forall x \forall y(((\text{Tree}(x) \wedge \text{Tree}(y)) \wedge (x = y)) \Rightarrow (\text{Size}(x) = \text{Size}(y)))$	from KB
L 15	$((\text{Tree}(x) \wedge \text{Tree}(y)) \wedge (x = y)) \Rightarrow (\text{Size}(x) = \text{Size}(y))$	open formula of L14
L 16	$\neg((\exists x)(\text{Tree}(x) \wedge (\text{Size}(x) = 1 + 1 + 1)))$	from $\neg\alpha$
L 17	$\forall x(\neg(\text{Tree}(x) \wedge (\text{Size}(x) = 1 + 1 + 1)))$	from L16 using exists negation
L 18	$\forall x(\neg\text{Tree}(x) \vee \neg(\text{Size}(x) = 1 + 1 + 1))$	from L17 using DeMorgan
L 19	$(\neg\text{Tree}(x) \vee \neg(\text{Size}(x) = 1 + 1 + 1))$	open formula of L18
L 20	Leaf(0)	CNF of L1
L 21	Leaf(1)	CNF of L2
L 22	$(\neg\text{Leaf}(x) \vee \text{Tree}(x))$	CNF of L4
L 23	$(\neg\text{Tree}(x) \vee \neg\text{Tree}(y) \vee \text{Tree}((x,y)))$	CNF of L6
L 24	$(\neg\text{Leaf}(x) \vee (\text{Size}(x) = 1))$	CNF of L8

L 25	$(\neg \text{Tree}(x) \vee \neg (\text{Size}(x) = f_1(x,y)) \vee \neg \text{Tree}(y) \vee \neg (\text{Size}(y) = f_2(x,y)) \vee (\text{Size}((x,y)) = f_1(x,y) + f_2(x,y)))$	CNF of L11
L 26	$(\neg(x = y) \vee \neg \text{Tree}(x) \vee \text{Tree}(y))$	CNF of L13
L 27	$(\neg \text{Tree}(x) \vee \neg \text{Tree}(y) \vee \neg(x = y) \vee (\text{Size}(x) = \text{Size}(y)))$	CNF of L15
L 28	$(\neg \text{Tree}(x) \vee \neg (\text{Size}(x) = 1 + 1 + 1))$	CNF of L19

CNF Clauses

1. Leaf(0)
2. Leaf(1)
3. $\neg \text{Leaf}(x) \vee \text{Tree}(x)$
4. $\neg \text{Tree}(x) \vee \neg \text{Tree}(y) \vee \text{Tree}((x,y))$
5. $\neg \text{Leaf}(x) \vee (\text{Size}(x) = 1)$
6. $\neg \text{Tree}(x) \vee \neg (\text{Size}(x) = f_1(x,y)) \vee \neg \text{Tree}(y) \vee \neg (\text{Size}(y) = f_2(x,y)) \vee (\text{Size}((x,y)) = f_1(x,y) + f_2(x,y))$
7. $\neg(x = y) \vee \neg \text{Tree}(x) \vee \text{Tree}(y)$
8. $\neg \text{Tree}(x) \vee \neg \text{Tree}(y) \vee \neg(x = y) \vee (\text{Size}(x) = \text{Size}(y))$
9. $\neg \text{Tree}(x) \vee \neg (\text{Size}(x) = 1 + 1 + 1)$

See included resolution_proof.pdf for the resolution refutation.

Samples of Unify algorithm use:

- I: Unify(Leaf(0), Leaf(x))
 - 1: Unify(Leaf(0), Leaf(x), ())
 - 2: Since Leaf(0) and Leaf(x) are terms, return (Unify(args(Leaf(0)), args(Leaf(x)), Unify(op(Leaf(0)), op(Leaf(x)), ())))
 - 3: Unify(0, x, Unify(Leaf(a), Leaf(a), ()))
 - 4: Unify(Leaf(a), Leaf(a), ())
 - 5: Since Leaf(a) == Leaf(a), return ()
 - 6: Since x is a variable, return (Unify-var(x, 0, ()))
 - 7: Unify-var(x, 0, ())
 - 8: Return (x ↦ 0)
 - 9: Return (x ↦ 0)
 - 10: Return (x ↦ 0)
- II: Unify(Tree((0,0)), Tree(x))
 - 1: Unify(Tree((0,0)), Tree(x), ())
 - 2: Since Tree((0,0)) and Tree(x) are terms, return (Unify(args(Tree((0,0))), args(Tree(x)), Unify(op(Tree((0,0))), op(Tree(x)), ())))

```

3: Unify((0,0), x, Unify(Tree(a), Tree(a), ()))
4: Unify(Tree(a), Tree(a), ())
5: Since Tree(a) == Tree(a), return ()
6: Since x is a variable, return (Unify-var(x, (0,0), ()))
7: Unify-var(x, (0,0), ())
8: Return (x ↦ (0,0))
9: Return (x ↦ (0,0))
10: Return (x ↦ (0,0))
III: Unify((Size(0) = 1), (Size(0) = f1((0,0))))
1: Unify((Size(0) = 1), (Size(0) = f1((0,0))), ())
2: Since (Size(0) = 1) and (Size(0) = f1((0,0))) are terms, return
(Unify(args((Size(0) = 1)), args((Size(0) = f1((0,0)))), Unify(op((Size(0) =
1)), op((Size(0) = f1((0,0)))), ())))
3: Unify((Size(0), 1), (Size(0), f1((0,0))), Unify((a = b), (a = b), ()))
4: Unify((a = b), (a = b), ())
5: Since (a = b) == (a = b), return ()
6: Since (Size(0), 1) and (Size(0), f1((0,0))) are lists, return
(Unify(tail((Size(0), 1)), tail((Size(0), f1((0,0)))),
Unify(head((Size(0), 1)), head((Size(0), f1((0,0)))), ())))
7: Unify(1, f1((0,0)), Unify(Size(0), Size(0), ()))
8: Unify(Size(0), Size(0), ())
9: Since Size(0) == Size(0), return ()
10: Since f1((0,0)) is a variable, return (Unify-var(f1((0,0)), 1,
()))
11: Unify-var(f1((0,0)), 1, ())
12: Return (f1((0,0)) ↦ 1)
13: Return (f1((0,0)) ↦ 1)
14: Return (f1((0,0)) ↦ 1)
15: Return (f1((0,0)) ↦ 1)

```

4.

Predicates

1. On(b, x) //checks if a block is on a particular block or on the table
2. LeftOf(b, x) //is block “b” on the left of block “x”
3. RightOf(b, x) //is block “b” on the left of block “x”
4. Block(x) //checks if a block is in fact a block
5. ClearLeft(y) //check if the left side of a block is clear
6. ClearRight(y) //check if the right side of a block is clear
7. ClearLeftTop(x) //checks if the top left of a box is clear

8. ClearRightTop(x)//checks if the top right of a box is clear
9. ClearLeftBottom(x)//checks if the bottom left of a box is clear
10. ClearRightBottom(x)//checks if the bottom right of a box is clear
11. Clear(x)//nothing is on x

Actions

12. MoveOntoTwo(b,w, x, y, z)

i. I imagine we need to have five parameters, the first being the block you want to move, the second and third being where it is, and the fourth and fifth being where you want the block to go

- Preconditions
 - //new one// $((\text{On}(b, \text{Table}) \wedge \text{ClearRightTop}(y) \wedge \text{ClearLeftTop}(z) \wedge \text{ClearRightTop}(b) \wedge \text{ClearLeftTop}(b)) \vee (\text{ClearLeftTop}(b) \wedge \text{ClearRightTop}(b) \wedge \text{OnLeft}(b, w) \wedge \text{OnRight}(b, x) \wedge \text{ClearRightTop}(y) \wedge \text{ClearLeftTop}(z) \wedge \text{LeftOf}(y, z) \wedge \text{RightOf}(z, y) \wedge \text{Block}(b) \wedge \text{Block}(w) \wedge \text{Block}(x) \wedge \text{Block}(y) \wedge \text{Block}(z)))$
- Effect \neg
 - $\text{OnRight}(b, y) \wedge \text{OnLeft}(b, z) \wedge \neg \text{OnLeftOf}(b, y) \wedge \neg \text{OnRightOf}(b, x) \wedge \neg \text{On}(b, \text{Table})$

MoveToTable(b, x, y)

- Preconditions
 - $\neg \text{OnTable}(b) \wedge \text{ClearLeft}(b) \wedge \text{ClearRight}(b) \wedge \text{Clear}(y) \wedge \text{Clear}(\text{Table})$
- Effect
 - $\text{On}(b, \text{Table}) \wedge \text{ClearRight}(x) \wedge \text{ClearLeft}(y)$

MoveOnto(b, x, w, y)

- Preconditions:
 -
- Effect:
 -

MoveLeftOf(b, x, w, y)

- Preconditions
 - $((\text{On}(b, \text{Table}) \wedge \text{ClearLeft}(y)) \vee (\text{ClearLeftTop}(b) \wedge \text{ClearRightTop}(b) \wedge \text{ClearLeft}(y)))$
 -

MoveRightOf(b, x, w, y)

- Preconditions
 - $((\text{On}(\text{b}, \text{Table}) \wedge \text{ClearRight}(\text{y})) \vee (\text{ClearLeftTop}(\text{b}) \wedge \text{ClearRightTop}(\text{b}) \wedge \text{ClearLeft}(\text{y})))$
 -

Solution

b

a b c \rightarrow **a c**

//this should basically move “b” on top of “c”

MoveOnto(b, b, c, c)

//this should basically move “a” on the left side of “c”

MoveLeftOf(a, c)

//this should move “b” onto “a” and “c”

MoveOntoTwo(b, b, a, c)

5.

PDDL air cargo description:

3 actions: *Load, Unload, Fly*

Action(Load(c, p, a),

PRECOND : At(c, a) \wedge At(p, a) \wedge Cargo(c) \wedge Plane(p) \wedge Airport(a)

EFFECT : \neg At(c, a) \wedge In(c, p))

Action(Unload(c, p, a),

PRECOND : In(c, p) \wedge At(p, a) \wedge Cargo(c) \wedge Plane(p) \wedge Airport(a)

EFFECT : At(c, a) \wedge \neg In(c, p))

Action(Fly(p, from, to),

PRECOND : At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)

EFFECT : \neg At(p, from) \wedge At(p, to))

Variables : a : Airport, c : Cargo, p : Plane, from : start, to : destination

Predicates :

1. *In(c, p)*

2. *At(c, a)*

3. *At(p, a)*

Frame Problem:

Notes reference in-class example:

Fly(P₁, SFO, SFO)

EFFECT according to fly rule : At(P₁, SFO) \wedge \neg At(P₁, SFO)

Result(s, a) = (s - Del(a) \cup Add(a))

from this we would remove *At(P₁, SFO)* and add *At(P₁, SFO)* which results in an anomaly.

Consider the action:

Load(C₁, P₂, SFO)